PTO 08-7333

CC=JP
DATE=19970516
KIND=A
PN=09128264

INFORMATION CONTROLLER AND CONTROL METHODS THEREOF

[Joho Seigyo Sochi Oyobi Sono Seigyo Hoho)

Takashi SAKAI, et al.

UNITED STATES PATENT AND TRADEMARK OFFICE

WASHINGTON, D.C. September 2008

TRANSLATED BY: SCHREIBER TRANSLATION, INC.

PUBLICATION COUNTRY	(10):	Japan
DOCUMENT NUMBER	(11):	09128264
DOCUMENT KIND	(12):	A
PUBLICATION DATE	(43):	19970516
APPLICATION NUMBER	(21):	07282963
APPLICATION DATE	(22):	19951031
INTERNATIONAL CLASSIFICATION	(51):	G06F 11/28
INVENTOR(S)	(72):	Takashi SAKAI, et al.
APPLICANT(S)	(71):	NEC IC Microcomputer System Co., Ltd.
TITLE	(54):	INFORMATION CONTROLLER AND CONTROL METHODS THEREOF
FOREIGN TITLE	(54A):	Johoigyo SeSochi Oyobi Sono Seigyo Hoho

/2[1]

[Claims]

[Claim 1] An information controller which has a tracing memory storing and holding predetermined trace information, a tracing counter holding a tracing memory ≅ address for storing said trace information in said tracing memory and incremented after said trace information is stored in said tracing memory every time an instruction of a predetermined program is executed, a rear branch detecting part detecting the presence or absence of rear branch occurrence by receiving a detection indication of a mode indication signal and comparing a preceding value and a present value of a predetermined program ≅ address, a branch destination register latching the value of a branch destination program ≅ address at the time of said rear branch occurrence, a branch origin program latching the value of a branch origin program ≅ address at the time of said rear branch occurrence, and a loop recognition part including a loop start register, a loop end register and a loop tracing counter for recognizing a repeated execution part of said program and

---

[1]Numbers in the margin indicate pagination in the foreign text.

stores only the first time and the final time of the repeated execution part of said program in said tracing memory

characterized in that

the information controller is constituted by having a CALL instruction detecting part detecting whether an executive instruction of said program is an instruction calling a subroutine or not, an RET instruction detecting part detecting whether an executive instruction of said program is an instruction returning from the subroutine, a call stack storing and holding multiple sets of respective values of the loop start register, loop end register and loop tracing counter, respectively,

a call stack counter counting and holding the setting number of respective values of the loop start register, loop end register and loop tracing counter stored in said call stack and,

if a subroutine call instruction exists in the repeated execution part of said program, only the first time and the final time of the repeated execution part of said program is stored in said tracing memory.

[Claim 2] An information control method of the information controller which has a tracing memory storing and holding predetermined trace information, a tracing counter holding a trace memory ≅ address for storing said trace information in said

tracing memory and incremented after said trace information is stored in said tracing memory every time an instruction of a predetermined program is executed, a rear branch detecting part detecting the presence or absence of rear branch occurrence by receiving a detection instruction of a mode indication signal and comparing a preceding value and a present value of a predetermined program ≅ address, a branch destination register latching the value of a branch destination program ≅ address at the time of said rear branch occurrence, a branch origin program latching the value of a branch origin program ≅ address at the time of said rear branch occurrence, and a loop recognition part including a loop start register, a loop end register and a loop tracing counter for recognizing a repeated execution part of said program and

stores only the first time and the final time of the repeated execution part of said program in said tracing memory

characterized by having

a first step wherein said loop start register, loop end register and loop tracing counter are initiated at the start of tracing, respectively,

a second step wherein contents of said branch destination register and contents of said loop start register as well as contents of said branch origin register and contents of said

loop end register are compared and collated at the time of detection of rear branch made by said rear branch detecting part after the start of tracing, respectively,

a third step wherein the value of said loop tracing counter is set up in said loop tracing counter when the contents of said branch destination register and contents of said loop start register as well as the contents of said branch origin register and contents of said loop end register are consistent in said second step,

a fourth step wherein the value of said branch destination register is set up in said loop start register, the value of said branch origin register is set up in said loop end register and the value of said loop tracing counter is set up in said loop tracing counter, respectively when the contents of said branch destination register and contents of said loop start register as well as the contents of said branch origin register and contents of said loop end register are inconsistent in said second step,

a fifth step wherein the respective values of said loop start register, loop end register and loop tracing counter are stored in positions assigned by the value of said call stack counter in said call stack when the execution of an instruction calling a subroutine is detected by said CALL instruction detecting part,

a sixth step wherein the value of said call stack counter is incremented,

a seventh step wherein said start register, said loop end register and said loop tracing counter are initiated, respectively,

an eighth step wherein the value of said call stack counter is decremented when the execution of an instruction returning from a subroutine is detected by said RET instruction detecting part, and

a ninth step wherein the values stored at the positions assigned by the values of said call stack counter in said call stack are set up in said loop start register, said loop end register and said loop tracing counter, respectively.

[Claim 3] An information control method of the information controller which has a tracing memory storing and holding predetermined trace information, a tracing counter holding a tracing memory ≅ address for storing said trace information in said tracing memory and incremented after said trace information is stored in said tracing memory every time an instruction of a predetermined program is executed, a rear branch detecting part detecting the presence or absence of rear branch occurrence by receiving a detection indication of a mode indication signal and comparing a preceding value and a present value of a

predetermined program ≅ address, a branch destination register latching the value of a branch destination program ≅ address at the time of said rear branch occurrence, a branch origin program latching the value of a branch origin program ≅ address at the time of said rear branch occurrence, and a loop recognition part including a loop start register, a loop end register and a loop tracing counter for recognizing a repeated execution part of said program and

stores only the first time and the final time of the repeated execution part of said program in said tracing memory

characterized by having

a first step wherein said loop start register, loop end register and loop tracing counter are initiated at the start of tracing, respectively,

a second step wherein contents of said branch destination register and contents of said loop start register as well as contents of said branch origin register and contents of said loop end register are compared and collated at the time of rear branch detection made by said rear branch detecting part after the start of tracing, respectively,

a third step wherein the value of said loop tracing counter is set up in said loop tracing counter when the contents of said

branch destination register and contents of said loop start register as well as the contents of said branch origin register and contents of said loop end register are consistent in said second step,

a fourth step wherein the value of said branch destination register is set up in said loop start register, the value of said branch origin register is set up in said loop end register and the value of said loop tracing counter is set up in said loop tracing counter, respectively when the contents of said branch destination register and contents of said loop start register as well as the contents of said branch origin register and contents of said loop end register are inconsistent in said second step,

a fifth step wherein the values of said call stack counter are increased by a portion of the total of respective sizes of said loop start register, loop end register and loop tracing counter by said CALL instruction detecting part at the time of detecting the execution of an instruction calling a subroutine, and

a sixth step wherein the values of said call stack counter are decreased by a portion of the total of respective sizes of said loop start register, loop end register and loop tracing counter by said RET instruction detecting part at the time of detecting the execution of an instruction returning from a subroutine.

[Detailed Description of the Invention]

[0001]

[Technical Field of the Invention] The present invention related to an information controller and control methods thereof.

[0002]

[Prior Art] A block diagram showing the constitution of one example of a prior information controller is shown in Fig. 3. This prior example is one example of an information controller disclosed on Japan Kokai 05-151023, and an operating procedure of tracing of said prior art is shown in a flow chart of Fig. 4.

[0003] As shown in Fig. 3, this prior example is constituted by having a tracing memory 1 storing trace information corresponding to a trace information bus 3 transmitting the trace information by a medium; a tracing counter 4 holding tracing memory ≅ address for storing the trace information inputted via the trace information bus 3 in the tracing memory 1 and incremented by a trace control signal 2 after said trace information is stored in the tracing memory 1; a rear branch detecting part 7 which includes a preceding value register 71 storing a preceding value of microprogram address 5 and a present value register 72 storing a present value of

microprogram address 5, compares the preceding value and the present value of microprogram address 5 when a detection indication is made by a mode indication signal 6 and detects the occurrence of rear branch when the preceding value is above the present value; a branch destination register 8 which holds the preceding value register 71 at the time of detecting the rear branch in the rear branch detecting part 7; a branch origin register 9 holding the present value register 72 at the time of detecting the rear branch in the rear branch detecting part 7; and loop recognition parts 10-1, 10-2, ⋯ 10-N which include loop start registers 101i (i = 1, 2, ⋯ , N), loop end registers 102i (i = 1, 2, ⋯ , N) and loop tracing counters 103i (i = 1, 2, ⋯ , N) and hold information for recognizing a repeated execution part of the microprogram. A head microprogram address of the repeated execution part of microprogram is stored in the loop start registers 101i, a final microprogram address of the repeating part of microprogram is stored in the loop end registers 102i and a value of tracing counter 4 at the time of recognizing the repeated part of microprogram is stored in the loop tracing counter 103-i in the above loop recognition parts. Loop identification IDs weighed, e.g., by size relation are given to these loop recognition parts, respectively.

[0004] Next, actions of this prior example are described with reference to Fig. 3 and Fig. 4. For the convenience of description, actions are described by supposing such a case that a loop recognition part is provided with only three loop recognition parts, i.e., the loop recognition part 10-1, the loop recognition part 10-2 and the recognition part 10-N (in this case, all other loop recognition parts 10-3, 10-4, ⋯ 10-(N-1) are taken outside the target of description of actions. However, the generality of description of actions of this prior example is not lost even if it is so supposed.)

/4

[0005] A trace action of this prior example are started by a trace control signal 2, the action is started at a time that the trace start is instructed by said trace control signal and the action is stopped at a time that the trace stop is instructed by said trace control signal. When the trace action is started, all loop start registers and loop end registers included inside all the loop recognition parts 10-1, 10-2, 10-N are initiated, e.g., at the maxima that can be set up in these registers (a step ST1). Subsequently, whether a rear branch is detected in the rear branch detecting part 7 or not is judged (a step ST2), if it is not detected, trace information is stored in a tracing memory 1 via the trace information bus 3, the tracing

counter 4 is incremented (a step ST7), the flow is returned to the step ST2 again and then processings from step T2 on are started again. If a rear branch is detected in the step T2, the value of loop end register 102-1 of loop recognition part 10-1 is judged to be below the value of branch destination register 8 or above the value of branch origin register 9 (a step ST3). If YES is judged in the step ST3, whether the value of loop start register 101-1 of loop recognition part 10-1 is equal to the value of branch destination register 8 and the value of loop end register 102-1 is equal to the value of branch origin register 9 or not is judged (a step ST4). If NO is judged in the step ST4, the value of branch destination register 8 is set up in the loop start register 101-1, the value of branch origin register 9 is set up in the loop end register 102-1 and the value of tracing counter 4 is set up in the loop tracing counter 103-1 of loop recognition part 10-1, loop recognition information held in the loop recognition part 10-1 is renewed (a step ST6), and the flow is returned to said step ST7. In the step ST7, as described above, the trace information is stored in the tracing memory 1 via the trace information bus 3, and the tracing counter 4 is incremented to perfrom processings from step T2 on again. If YES is judged in said step ST4, the value of loop tracing counter 102-1 of loop recognition part 10-1 is set up in the tracing

counter 4 (a step ST5), subsequently the flow is returned to said ST7. In said step ST7, the trace information is stored in the tracing memory 1, and the tracing counter 4 is incremented to perform processings from step T2 on again.

[0006] On the other hand, if NO is judged in the step ST3, whether the value of loop tracing counter 102-2 of loop recognition part 10-2 is judged to be below the value of branch destination register 8 or above the value of branch origin register 9 (a step ST10). If YES is judged in the step ST3, whether the value of loop start register 101-2 (wrong number "102-2" in original document, translator) of loop recognition part 10-2 is equal to the value of branch destination register 8 and the value of loop end register (wrong name "loop start register" in original document, translator) 102-2 is equal to the value of branch origin register 9 or not is judged (a step ST11). If NO is judged in the step S11, the value of branch destination register 8 is set up in the loop start register 101-2, the value of branch origin register 9 is set up in the loop end register 103-2 and the value of tracing counter 4 is set up in the loop tracing counter 103-2 of loop recognition part 10-2 to renew loop recognition information held in the loop recognition part 10-2 (a step ST13). Subsequently, the initiation of loop recognition part 10-1 is performed, e.g., by

setting maxima that can be set for the loop start register 101-1 and the loop end register 102-1 of loop recognition part 10-1 in said registers (a step ST14), subsequently the flow is returned to said step ST7. In the step ST7, the trace information is stored in the tracing memory 1, and the tracing counter 4 is incremented to perform processings from step T2 on again.

[0007] If YES is judged in the step S11, the value of loop tracing counter 103-2 of loop recognition part 10-2 is set up in the tracing counter 4 (a step ST12), the flow is returned to said ST7, the trace information is stored in the tracing memory 1, and the tracing counter 4 is incremented to perform processings from step ST2 on again. If NO is judged in said step ST10, whether the value of loop trance counter 102-N of loop recognition part 10-N is judged to be below the value of branch destination register 8 or above the value of branch origin register 9 (a step ST20); if YES is judged, whether the value of loop start register 101-N of loop recognition part 10-N is equal to the value of branch destination register 8 and the value of loop end register 102-N is equal to the value of branch origin register 9 or not is judged (a step ST21). If NO is judged in the step ST21, the value of branch destination register 8 is set up in the loop start register 101-N, the value of branch origin register 9 is set up in the loop end register 102-N and the

value of tracing counter 4 is set in the loop tracing counter 103-N of loop recognition part 10-N to renew the loop recognition information held in the loop recognition part 10-N (a step ST23).

[0008] Subsequently, the initiation of loop recognition parts 10-1 and 10-2 is performed, e.g., by setting maxima for the loop start registers 101-1, 101-2 and the loop end register 102-1, 102-2 of loop recognition parts 10-1 and 10-2 (a step ST24), the flow is returned to said step ST7, the trace information is stored in the tracing memory 1, and the trace

/5

counter 4 is incremented to perform processings from step ST2 on again. If YES is judged in the step ST21, the value of loop tracing counter 103-N of loop recognition part 10-N is set up in the tracing counter 4 (a step ST22), similarly, the flow is returned to said step ST7, trace information is stored in the tracing memory 1, and the tracing counter 4 is incremented to perform processings from step ST2 on again. This is same as the case of being judged to be NO, the flow is returned to said step ST7, trace information is stored in the tracing memory 1, and the tracing counter 4 is incremented to perform processings from step ST2 on again.

[0009]

[Problem to Be Solved by the Invention] In the above-mentioned prior information controller and control methods thereof, the collection and control of tracing is carried out with the information detected in the rear branch detecting part, therefore, when a routine call exists in the repeated execution part of microprogram and accordingly the repeated execution part further exists in a called child routine, it becomes an event that loop recognition information of the subroutine is registered in the loop recognition part and the loop recognition information of a parent routine is thrown away from the loop recognition part simultaneously at a time that a rear branch occurs in said subroutine by making it in conformity with a condition that the head microprogram address of repeated execution part of said child routine is above the end microprogram address of repeated execution part of said parent routine calling the child routine or the end microprogram address of repeated execution part of said child routine is below the end microprogram address of repeated execution part of said parent routine.

[0010] As a result, the repeated execution part of said subroutine is traced only in the first time and the final time, but the repeated execution part of said parent routine is traced

in all the repeated times. Thus, when a subroutine call exists in the prior information controller and control methods thereof, there is such a disadvantage that trace information cannot be collected with good efficiency.

[0011] The purpose of present invention consists in solving the above problem and realizing an information controller and control methods thereof which enable to recognize a subroutine control part in a microprogram by detecting a subroutine call instruction and a return instruction from the subroutine and collect trace information with good efficiency.

[0012]

[Problems to Be Solved by the Invention] An information controller of the first invention is an information controller which has a tracing memory storing and holding predetermined trace information, a tracing counter holding a tracing memory ≅ address for storing said trace information in said tracing memory and incremented after said trace information is stored in said tracing memory every time an instruction of a predetermined program is executed, a rear branch detecting part detecting the presence or absence of rear branch occurrence by receiving a detection indication of a mode indication signal and comparing a preceding value and a present value of a predetermined program ≅ address, a branch destination register latching the value of a

branch destination program ≅ address at the time of said rear branch occurrence, a branch origin program latching the value of a branch origin program ≅ address at the time of said rear branch occurrence, and a loop recognition part including a loop start register, a loop end register and a loop tracing counter for recognizing a repeated execution part of said program and

stores only the first time and the final time of the repeated execution part of said program in said tracing memory

characterized in that

the information controller is constituted by having a CALL instruction detecting part detecting whether the execution of said microprogram is an instruction calling a subroutine or not, an RET instruction detecting part detecting whether the executive instruction of said program is an instruction returning from the subroutine, a call stack storing and holding multiple sets of respective values of the loop end register and the loop tracing counter, respectively,

a call stack counter counting and holding the setting number of respective values of the loop start register, loop end register and loop tracing counter stored in said call stack and, when a subroutine call instruction exists in the repeated execution part of said program, only the first time and the final time of

the repeated execution part of said program is stored in said tracing memory.

[0013] An information control method of the second invention is an information control method of the information controller which has a tracing memory storing and holding predetermined trace information, a tracing counter holding a tracing memory ≅ address for storing said trace information in said tracing memory and incremented after said trace information is stored in said tracing memory every time an instruction of a predetermined program is executed, a rear branch detecting part detecting the presence or absence of rear branch occurrence by receiving a detection indication of a mode indication signal and comparing a preceding value and a present value of a predetermined program ≅ address, a branch destination register latching the value of a branch destination program ≅ address at the time of said rear branch occurrence, a branch origin program latching the value of a branch origin program ≅ address at the time of said rear branch occurrence, and a loop recognition part including a loop start register, a loop end register and a loop tracing counter for recognizing a repeated execution part of said program and

/6

stores only the first time and the final time of the repeated execution part of said program in said tracing memory

characterized by having

a first step wherein said loop start register, loop end register and loop tracing counter are initiated at the start of tracing, respectively,

a second step wherein contents of said branch destination register and contents of said loop start register as well as contents of said branch origin register and contents of said loop end register are compared and collated at the time of detection of rear branch made by said rear branch detecting part after the start of tracing, respectively,

a third step wherein the value of said loop tracing counter is set up in said loop tracing counter when the contents of said branch destination register and contents of said loop start register as well as the contents of said branch origin register and contents of said loop end register are consistent in said second step,

a fourth step wherein the value of said branch destination register is set up in said loop start register, the value of said branch origin register is set up in said loop end register and the value of said loop tracing counter is set up in said loop tracing counter, respectively when the contents of said

branch destination register and contents of said loop start register as well as the contents of said branch origin register and contents of said loop end register are inconsistent in said second step,

a fifth step wherein the respective values of said loop start register, loop end register and loop tracing counter are stored in positions assigned by the value of said call stack counter in said call stack when the execution of an instruction calling a subroutine is detected by said CALL instruction detecting part,

a sixth step wherein the value of said call stack counter is incremented,

a seventh step wherein said start register, said loop end register and said loop tracing counter are initiated, respectively,

an eighth step wherein the value of said call stack counter is decremented when the execution of an instruction returning from a subroutine is detected by said RET instruction detecting part, and

a ninth step wherein the values stored at the positions assigned by the values of said call stack counter in said call stack are set up in said loop start register, said loop end register and said loop tracing counter, respectively.

[0014] An information control method of the second invention is an information control method which has a tracing memory storing and holding predetermined trace information, a tracing counter holding a tracing memory ≅ address for storing said trace information in said tracing memory and incremented after said trace information is stored in said tracing memory every time an instruction of a predetermined program is executed, a rear branch detecting part detecting the presence or absence of rear branch occurrence by receiving a detection indication of a mode indication signal and comparing a preceding value and a present value of a predetermined program ≅ address, a branch destination register latching the value of a branch destination program ≅ address at the time of said rear branch occurrence, a branch origin program latching the value of a branch origin program ≅ address at the time of said rear branch occurrence, and a loop recognition part including a loop start register, a loop end register and a loop tracing counter for recognizing a repeated execution part of said program and

stores only the first time and the final time of the repeated execution part of said program in said tracing memory

characterized by having

a first step wherein said loop start register, loop end register and loop tracing counter are initiated at the start of tracing, respectively,

a second step wherein contents of said branch destination register and contents of said loop start register as well as contents of said branch origin register and contents of said loop end register are compared and collated at the time of rear branch detection made by said rear branch detecting part after the start of tracing, respectively,

a third step wherein the value of said loop tracing counter is set up in said loop tracing counter when the contents of said branch destination register and contents of said loop start register as well as the contents of said branch origin register and contents of said loop end register are consistent in said second step,

a fourth step wherein the value of said branch destination register is set up in said loop start register, the value of said branch origin register is set up in said loop end register and the value of said loop tracing counter is set up in said loop tracing counter, respectively when the contents of said branch destination register and contents of said loop start register as well as the contents of said branch origin register

and contents of said loop end register are inconsistent in said second step,

a fifth step wherein the values of said call stack counter are increased by a portion of the total of respective sizes of said loop start register, loop end register and loop tracing counter by said CALL instruction detecting part at the time of detecting the execution of an instruction calling a subroutine, and

a sixth step wherein the values of said call stack counter are decreased by a portion of the total of respective sizes of said loop start register, loop end register and loop tracing counter by said RET instruction detecting part at the time of detecting the execution of an instruction returning from a subroutine.

[0015]

[Embodiment of the Invention] Next, the present invention is described with reference to drawings.

[0016] Fig. 1 is a block diagram showing the constitution of one embodiment of the present invention. As shown in Fig. 1, this embodiment is constituted by having a tracing memory 1 storing trace information corresponding to a trace information bus 3 for transmitting the trace information by a medium; a tracing counter 4 holding a tracing memory address for storing the trace information inputted via the trace information bus 3 in the tracing memory 1, storing said trace information in the

tracing memory 1 and then incremented with a trace control signal 2; a rear branch detecting part 7 including a preceding value register 71 for storing a preceding value of a microprogram address 5 and a present value register 72 for storing a present value of the microprogram address 5, comparing the preceding value and the present value of microprogram address 5 when they are detected and indicated with a mode

/7

indication signal 6 and detecting that a rear branch occurs if the preceding value is above the present value; a branch destination register 8 holding the preceding value register 71 at the time of detecting the rear branch in said rear branch detecting part 7; a branch origin register 9 holding the present value register 72 at the time of detecting the rear branch in said rear branch detecting part 7; loop recognition parts 10-1, 10-2, 10-N including loop start registers 101-i (i = 1, 2, ···, N), loop end registers 102-i (i = 1, 2, ···, N) and loop tracing counters 103-i (i = 1, 2, ···, N) and holding information for recognizing a repeated execution part of the microprogram, respectively; a CALL instruction detecting part 21 detecting whether an execution instruction of microprogram is an instruction calling a subroutine or not; an RET instruction detecting part 22 detecting whether the execution instruction of

microprogram is an instruction for returning from a subroutine; a call stack 23 which is a FILO type region used for fetching loop recognition information stored in these loop recognition parts; and a call stack counter 24 showing a number of loop recognition information stored in said call stack 23.

[0017] Like the case of prior example, the head microprogram address of the repeating part of microprogram is stored in the loop start registers 101i, the end microprogram address of the repeating part of microprogram is stored in the loop end registers 102i and the value of tracing counter 4 at the time of recongnizing the repeating part of microprogram is stored in the loop tracing counters 103-i in the above loop recognition parts. Loop identification IDs weighed, e.g., by size relation are given to these loop recognition parts, respectively.

[0018] Fig. 2 is a flow chart showing an operating procedure of this embodiment. As is evident by a contrast to Fig. 4, in this embodiment, steps including step ST31, step ST32, step ST33, step ST34, step ST35 and step ST36 are newly added as processing steps associated with addition of the constitution of devices including the CALL instruction detecting part 21, RET instruction detecting part 22, call stack 23, call stack counter 24, etc.

[0019] Next, actions of this embodiment (wrong words “prior example” in original document, translator) are described with reference to Fig. 1 and Fig. 2. Like the case of prior example, actions are described by supposing a case of providing only three loop recognition parts, i.e., a loop recognition part 10-1, a loop recognition part 10-2 and a loop recognition part 10-N described in Fig. 1 for the convenience of description.

[0020] At the start of tracing action, all of loop start registers and loop end registers included inside the loop recognition parts are initiated (step ST31). Next, the call stack counter 24 is initiated to, e.g., 0, whether the execution of a routine call instruction is detected or not in the CALL instruction detecting part 21 is judged in the tracing action step ST32); if the execution of a routine call instruction is detected, loop recognition information including values of loop start registers 101-1, 101-2, 101-N, values of loop end registers 102-1, 102-2, 102-N and values of loop tracing counter parts 103-1, 103-2, 103-N held in the loop recognition parts 10-1, 10-2, 10-N is stored in a predetermined position in the call stack 23 assigned by the value of call stack counter 24, and the call stack counter 24 is incremented (step ST33). Then, the loop recognition information in loop recognition parts 10-1, 10-2, 10-N are initiated (step ST34), trace information is stored in

the tracing memory 1 via the trace information bus 3, the tracing counter 4 is incremented (step ST7), and the flow is returned to the step ST2 and actions from step ST2 on are started again. If the execution of a subroutine call instruction is not detected in the step ST32, whether the execution of a subroutine return instruction is detected in the RET instruction detecting part 22 or not is judged in the trace action (step ST35). If the execution of a subroutine return instruction is detected, the call stack counter 24 is incremented, a value stored in a predetermined position in the call stack 23 assigned by the call stack counter 24 is fetched and set up as loop recognition information in the loop recognition parts 10-1, 10-2, 10-N (step ST36). If the execution of a subroutine return instruction is not detected in the step ST35, whether a rear branch is detected in the rear branch detecting part 7 or not is judged (step ST2). If a rear branch is not detected, trace information is stored in the tracing memory 1 via the trace information bus 3 and the tracing counter 4 is incremented (step ST7), then the flow is returned to the step ST2 again and actions from the step ST2 on are started. If a rear branch is detected in the step ST2, whether the value of loop end register 102-1 of the loop recognition part 10-1 is below the value of

branch destination register 8 or above the value of branch origin register 9 is judged (step ST3).

[0021] If YES is judged in the step ST3, whether the value of loop start register 101-1 of the loop recognition part 10-1 is equal to the value of branch destination register 8 and the value of loop end register 102-1 is equal to the value of branch

/8

origin register 9 or not is judged (step ST4). If NO is judged in the step ST4, the value of branch destination register 8 is set up in the loop start register 101-1 of loop recognition part 10-1, the value of branch origin register 9 is set up in the loop end register 102-1, the value of tracing counter 4 is set up in the loop tracing counter 103-1 to renew the loop recognition information held in the loop recognition part 10-1 (step ST6), then the flow is returned to said step ST7. In the step ST7, as described above, the trace information is stored in the tracing memory 1 via the trace information bus 3, and the tracing counter 4 is incremented to perform the processings from the step ST2 on again. If YES is judged in the step ST4, the value of loop tracing counter 103-1 of loop recognition part 10-1 is set up in the tracing counter 4 (step ST5), then the flow is returned to said step ST7. In the step ST7, the trace information is stored in the tracing memory 1 via the trace

information bus 3, and the tracing counter 4 is incremented to perform the processings from the step ST2 on again.

[0022] On the other hand, if NO is judged to be in the step ST3, whether the value of loop end register 102-1 of the loop recognition part 10-2 is below the value of branch destination register 8 or above the value of branch origin register 9 is judged (step ST10). If YES is judged, whether the value of loop start register 101-2 (wrong number “102-2” in original document, translator) of the loop recognition part 10-2 is equal to the value of branch destination register 8 and the value of loop end register 102-2 of the loop recognition part 10-2 is equal to the value of branch origin register 9 or not is judged (step ST11). If NO is judged in step ST11, the value of branch destination register 8 is set up in the loop start register 101-2, the value of branch origin register 9 is set up in the loop end register 102-2 (wrong number “103-2 in original document, translator) and the value of tracing counter 4 is set up in the loop tracing counter 103-2 of loop recognition part 10-2 to renew the loop recognition information held in the loop recognition part 10-2 (step ST13). Next, the initiation of loop recognition part 10-2 is performed, e.g., by setting maxima that can be set up in said register for the loop start register 101-1 and loop end register 102-1 of the loop recognition part 10-1 (step ST14),

subsequently the flow is returned to said step ST7. In step ST7, the trace information is stored in the tracing memory 1, and the tracing counter 4 is incremented to perform processings from step ST2 on again.

[0023] If YES is judged in the step ST11, the value of loop tracing counter 103-2 of loop recognition part 10-2 is set up in the tracing counter 4 (step ST12), the flow is returned to said step ST7, the trace information is stored in the tracing memory 1, and the tracing counter 4 is incremented to perform the processings from step ST2 on again. If NO is judged in the step ST10, whether the value of loop end register 102-N of loop recognition part 10-N is below the value of branch destination register 8 or above the value of branch origin register 9 is judged (step ST20). If YES is judged in the step ST20, whether the value of loop start register 101-N of loop recognition part 10-N is equal to the value of branch destination register 8 and the value of loop end register 102-N of loop recognition part 10-N is equal to the value of branch origin register 9 or not is judged (step ST21). If NO is judged in step 21, the value of branch destination register 8 is set up in the loop start register 101-N of loop recognition part 10-N, the value of branch origin register 9 is set up in the loop end register 102-N, the value of tracing counter 4 is set up in the loop tracing

counter 103-N to renew the loop recognition information held in the loop recognition part 10-N (step ST23).

[0024] Next, the loop recognition parts 10-1 and 10-2 are initiated by setting up, e.g., maxima for the loop start registers 101-1, 101-2 and loop end registers 102-1, 102-2 in the loop recognition part 10-1 and 10-2 (step 24), the flow is returned to ST7, the trace information is stored in the tracing memory 1, and the tracing counter 4 is incremented to perform the processings from step ST2 on. If YES is judged in step ST21, the value of loop tracing counter 103-N of loop recognition part 10-N is set up in the tracing counter 4 (step ST22), similarly, the flow is returned to said step ST7, the tracing information is stored in the tracing memory 1, and the tracing counter 4 is incremented to perform the processings from step ST2 on. This is same as the case that NO is judged in the step ST20, the flow is returned to said step ST7, the tracing information is stored in the tracing memory 1, and the tracing counter 4 is incremented to perform the processings from step ST2 on.

[0025] As described above, the loop recognition information held in the loop recognition part can be switched between the parent routine calling a subroutine and the called child routine, therefore tracing actions can also be performed with

good efficiency in a microprogram including a subroutine. Namely, the present invention enables to realize an information controller capable of recognizing a subroutine control part in the microprogram by detection of a subroutine call instruction and a return instruction from the subroutine and collecting trace information with good efficiency and realize control

/9

methods thereof.

[0026] Such an information control method wherein the call stack counter counts and holds the set number of respective values of loop start register, loop end register and loop tracing counter stored in the call stack, when the execution of subroutine call instruction is detected in the CALL instruction detecting part during the tracing action, the loop recognition information held in all loop recognition parts is stored in the call stack and then the call stack counter is incremented; when the execution of subroutine return instruction is detected in the RET instruction detecting part during the tracing action, the call stack counter is incremented, then the loop recognition information stored in the call stack is fetched and set up in all the loop recognition parts was mentioned in the above description, but an information control method different from this method wherein the call stack is mounted in a memory, when

the call stack counter holds an address of said call stack, the execution of a subroutine call instruction is detected in the CALL instruction detecting part during the tracing action, the loop recognition information held in all loop recognition parts is stored in the call stack, then the value of call stack counter is increased by a portion of the total of respective sizes of the loop start register, loop end register and loop tracing counter; when the execution of a subroutine return instruction is detected in the RET instruction detecting part during the tracing action, the value of call stack counter is decreased by a portion of the total of respective sizes of the loop start register, loop end register and loop tracing counter is set up in all the loop recognition parts may also be used. “What are effects of the present invention?” is specifically described below with reference to one application example of microprogram.

[0027] Fig. 5(a) is one example of microprogram including a subroutine. Fig. 5(b) is a chart showing a result obtained from tracing the microprogram shown in the above Fig. 5(a) by the prior information controller, and Fig. 5(c) is a chart showing a result obtained from tracing the microprogram shown in the above Fig. 5(a) by the present invention. In the prior information controller and its control method, the trace information from

second time to (N-1)th time of repeated processings (A) and (B) of parent routine and the trace information of first time and the last time of repeated processings (C) of the subroutine remain in the tracing memory as shown in Fig. 5(b), while said trace information are improved so that they do not remain in the tracing memory as shown in Fig. 5(c). If the consumption of tracing memory in the processing A, processing B and processing C of microprogram shown in Fig. 5(a) is assumed to be totally 500 bytes and repetition numbers N and M are assumed to be totally 10, the consumption of tracing memory in the prior information controller and its control methods is 16,000 bytes, while the consumption of tracing memory in the present invention is stopped to 4,000 bytes, thus the use efficiency is improved by a factor of 4. Accordingly, 4-fold execution result of microprogram can be remained in said tracing memory, the debugging range of microprogram based on trace data is enlarged to 4 times, thus the debugging efficiency can be improved by the tracing memory of same capacity.

[0028]

[Effects of the Invention] As described above, the present invention has such effects that the execution of a subroutine call instruction and the execution of a return instruction are detected by having a CALL instruction detecting part and an RET

instruction detecting part, when loop recognition information of a loop recognition part used until the time of detection is held in a call stack in the detection of said subroutine instructions, the loop recognition information held in the call stack is returned to the loop recognition part by a preceding subroutine call in the detection of a subroutine return instruction, even when a subroutine call is not included in a repeated execution part of microprogram by switching the loop recognition information held in the loop recognition part between a parent routine and a child routine to perform a tracing action, trace information is remained only for the first time and the final time of the repeated execution part, an even broader range of execution result of microprogram can be remained in a tracing memory of microprogram including a subroutine by a tracing memory of same capacity, thereby, the range of microprogram capable of debugging is enlarged and the debugging efficiency can be enhanced.

[Brief Description of the Drawings]

[Fig. 1] Block diagram showing the constitution of one embodiment of the present invention.

[Fig. 2] Drawing showing a flow chart of operating procedure in said embodiment.

[Fig. 3] Block diagram showing the constitution of prior example.

[Fig. 4] Drawing showing a flow chart of operating procedure in said prior example.

[Fig. 5] Drawing showing a microprogram and tracing data of the microprogram.

[Description of the Symbols]

1 tracing counter

2 trace control signal

3 trace information bus

4 tracing counter

5 microprogram address

6 mode indication signal

/10

7 rear branch detecting part

8 branch destination register

9 branch origin register

10-1, 10-2, JJJ, 10-N loop recognition parts

21 CALL instruction detecting part

22 RET instruction detecting part

23 call stack

24 call stack counter

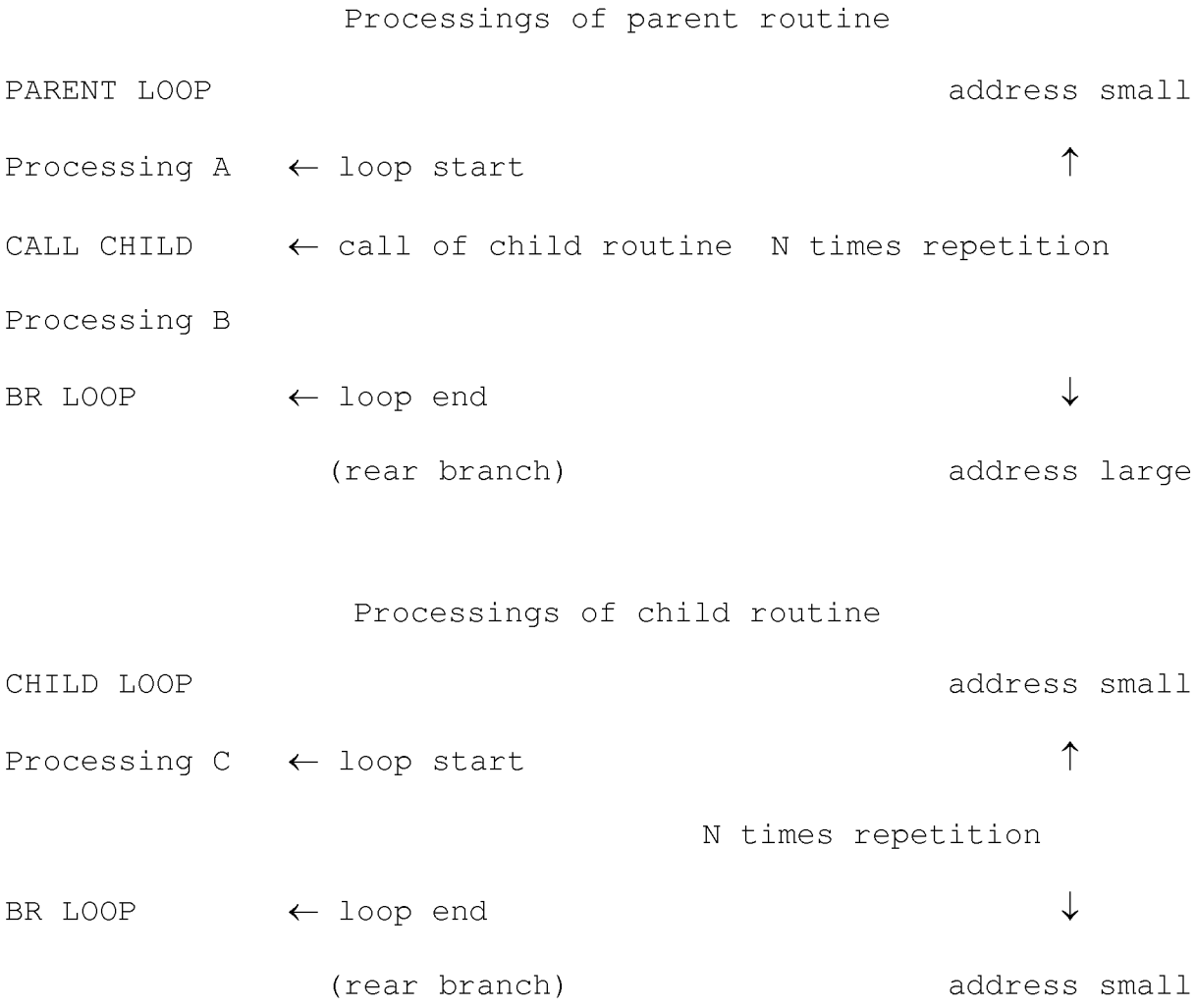
101-1, 101-2, ···, 101-N loop start registers

102-1, 102-2, ···, 102-N loop end registers

103-1, 103-2, ···, 103-N loop tracing counters

ST1 ~ ST24, ST31 ~ ST36 steps

[Fig. 5]



(a)

Proc. A (1st)	Proc. C (1st)	Proc. C (Mth)	Proc. B (1st)	Proc. A (2nd)	Proc. C (1st)	Proc. C (Mth)	Proc. B (2nd)	JJ
JJ	Proc. A ((N-1)th)	Proc. C (1st)	Proc. C (Mth)	Proc. B ((N-1)th)	Proc. A (Nth)	Proc. C (1st)	Proc. C (Mth)	Proc. B (Nth)

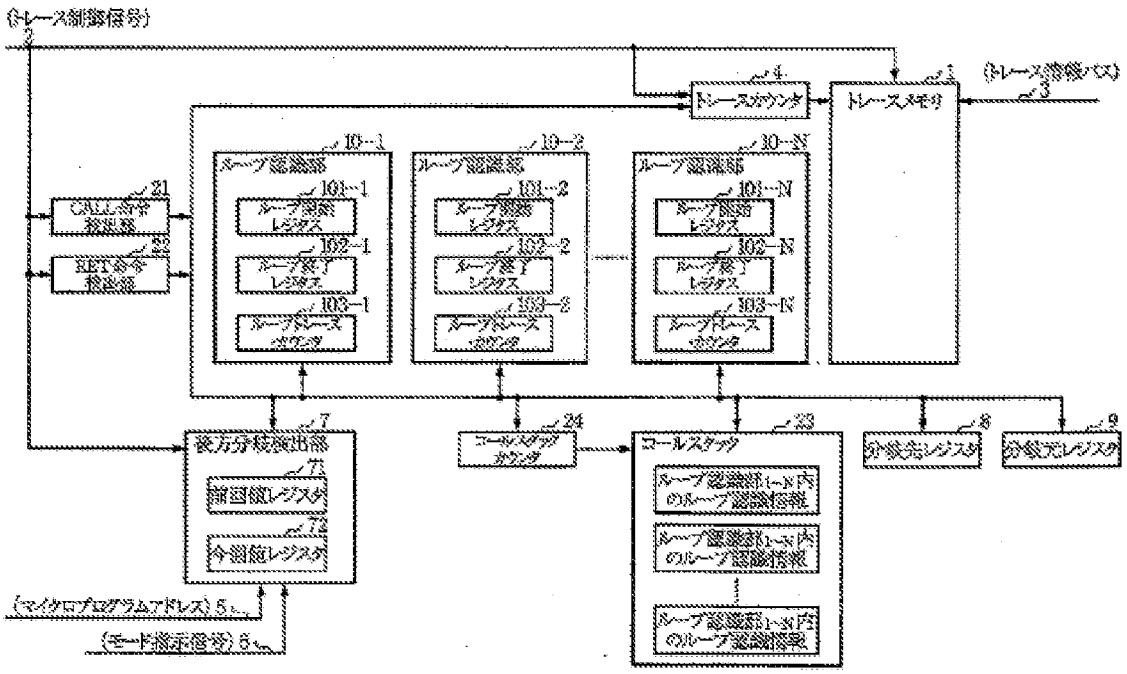
(b)

Proc. A (1st)	Proc. C (1st)	Proc. C (Mth)	Proc. B (1st)	Proc. A (Nth)	Proc. C (1st)	Proc. C (Mth)	Proc. B (Nth)

(c)

/11

[Fig. 1]



1 tracing memory

2 trace control signal

3 trace information bus

4 tracing counter

5 microprogram address

6 mode indication signal

7 rear branch detecting part

71 preceding value register

72 present value register

8 branch destination register

9 branch origin register

10-1 loop recognition part

101-1 loop start register

102-1 loop end register

103-1 loop tracing counter

10-2 loop recognition part

101-2 loop start register

102-2 loop end register

103-2 loop tracing counter

10-N loop recognition part

101-N loop start register

102-N loop end register

103-N loop tracing counter

21 CALL instruction detecting part

22 RET instruction detecting part

23 call stack

loop recognition information in loop recognition$_{1\sim N}$

loop recognition information in loop recognition$_{1\sim N}$

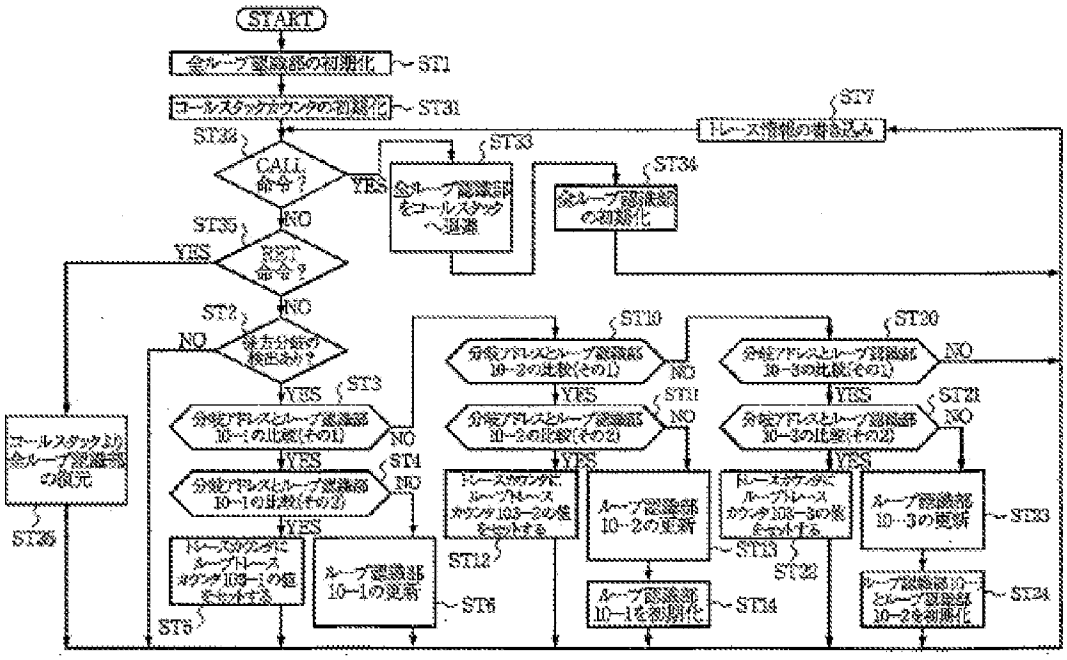
:
:

loop recognition informaation in loop recognition$_{1\sim N}$

24 call stack counter

/12

[Fig. 2]



START

ST1 Initiation of all loop recognition parts

ST2 Does detection of a rear branch exist?

ST3 Comparison of branch address and loop recognition part 10-1 (1)

ST4 Comparison of branch address and loop recognition part 10-1 (2)

ST5 Set value of loop tracing counter 103-1 in tracing counter

ST6 Renewal of loop recognition part 10-1

ST7 Writing of trace information

ST8 Branch destination register

ST9 Branch origin register

ST10 Comparison of branch address and loop recognition part 10-2 (1)

ST11 Comparison of branch address and loop recognition part 10-2 (2)

ST12 Comparison of branch address and loop recognition part 103-2 (2)

ST13 Renewal of loop recognition part 10-2

ST14 Initiation of loop recognition part 10-1

ST20 Comparison of branch address and loop recognition part 10-3 (1)

ST21 Comparison of branch address and loop recognition part 10-3 (2)

ST22 Set value of loop tracing counter 103-3

ST23 Renewal of loop recognition part 10-3

ST24 Initiation of loop recognition part 10-1 and loop recognition part 10-2

ST31 Initiation of call stack counter

ST32 Is CALL instruction?

ST33 Evacuate all loop recognition parts to call stack

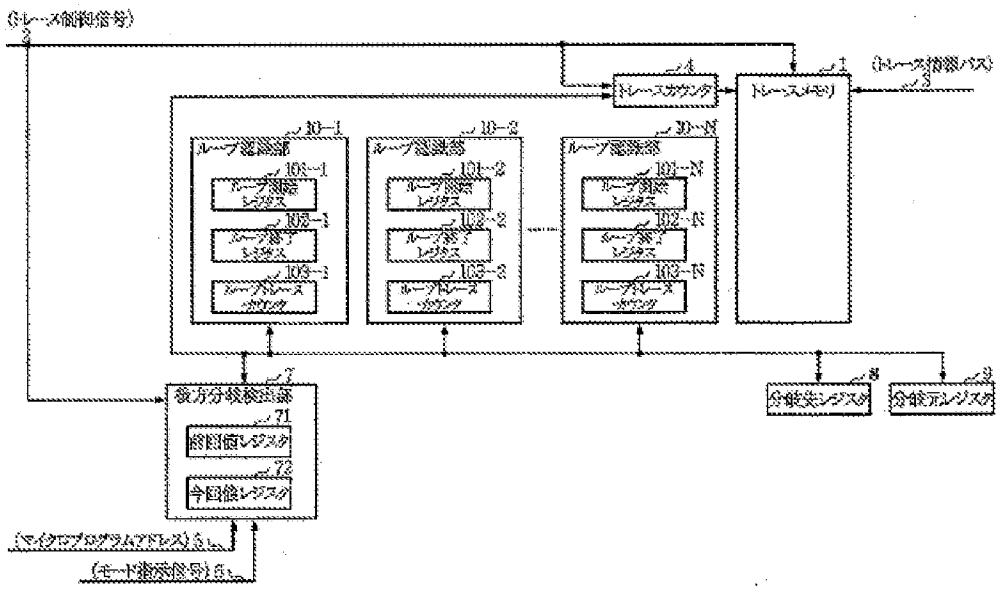
ST34 Initiation of loop recognition parts

ST35 Is RET instruction?

ST36 Restoration of all loop recognition parts from call stack

/13

[Fig. 3]



1 tracing memory

2 trace control signal

3 trace information bus

4 tracing counter

5 microprogram address

6 mode indication signal

7 rear branch detecting part

71 preceding value register

72 present value register

8 branch destination register

9 branch origin register

10-1 loop recognition part

101-1 loop start register

102-1 loop end register

103-1 loop tracing counter

10-2 loop recognition part

101-2 loop start register

102-2 loop end register

103-2 loop tracing counter

10-N loop recognition part

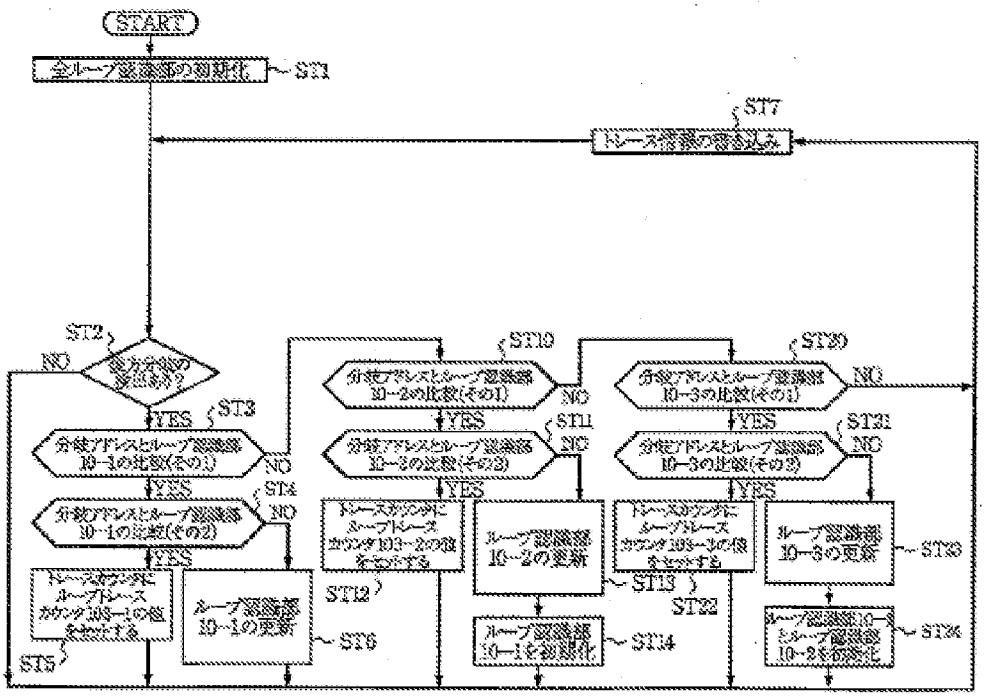
101-N loop start register

102-N loop end register

103-N loop tracing counter

/14

[Fig. 4]



START

ST1 Initiation of all loop recognition parts

ST2 Does detection of a rear branch exist?

ST3 Comparison of branch address and loop recognition part 10-1 (1)

ST4 Comparison of branch address and loop recognition part 10-1 (2)

ST5 Set value of loop tracing counter 103-1 in tracing counter

ST6 Renewal of loop recognition part 10-1

ST7 Writing of trace information

ST8 Branch destination register

ST9 Branch origin register

ST10 Comparison of branch address and loop recognition part 10-2 (1)

ST11 Comparison of branch address and loop recognition part 10-2 (2)

ST12 Comparison of branch address and loop recognition part 103-2 (2)

ST13 Renewal of loop recognition part 10-2

ST14 Initiation of loop recognition part 10-1

ST20 Comparison of branch address and loop recognition part 10-3 (1)

ST21 Comparison of branch address and loop recognition part 10-3 (2)

ST22 Set value of loop tracing counter 103-3

ST23 Renewal of loop recognition part 10-3

ST24 Initiation of loop recognition part 10-1 and loop recognition part 10-2